Script Systems and Frameworks

Gordon Watts

4-28-97

Big Time C++ Review

- Scripting Systems in D0
- Framework Processing in D0

You can find all of this talk on WWW:

http://d0sgi0.fnal.gov/gwatts/talks/

Scripting Systems in DØ

Not a shell script

- Used as part of the program.
- Objects accessed by the script system directly.

Two Environments:

- Online
- Offline

Online

Why?

- Run 1 lacked integration
  - Separate programs to do separate things.
  - Use to stitch together the system.
  - No central control possible.
  - Shifter had to memorize just about everything.
  - Not fast, mistakes...
- GUI (tkinter).
- On the fly scripts can be written by the shifters.

Offline

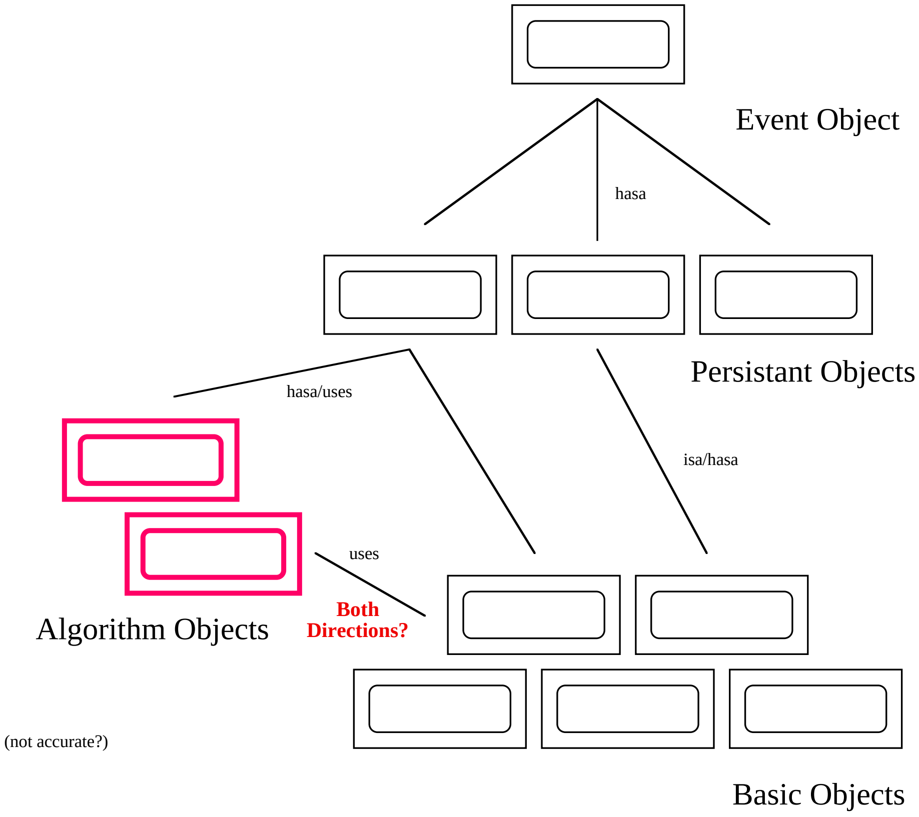
- Normalize the various input scripts (RCP, PB, etc.).
- GUI.
- Flexible way to put together system.
- Remote control (batch environment).

Depends upon how the data model turns out!

- Assembly line reconstruction.
  - Algorithm control is at a high level.
  - Use scripts as flexible glue.
- On demand reconstruction.
  - Algorithm called by the data.
  - Doesn't make nearly as much sense.

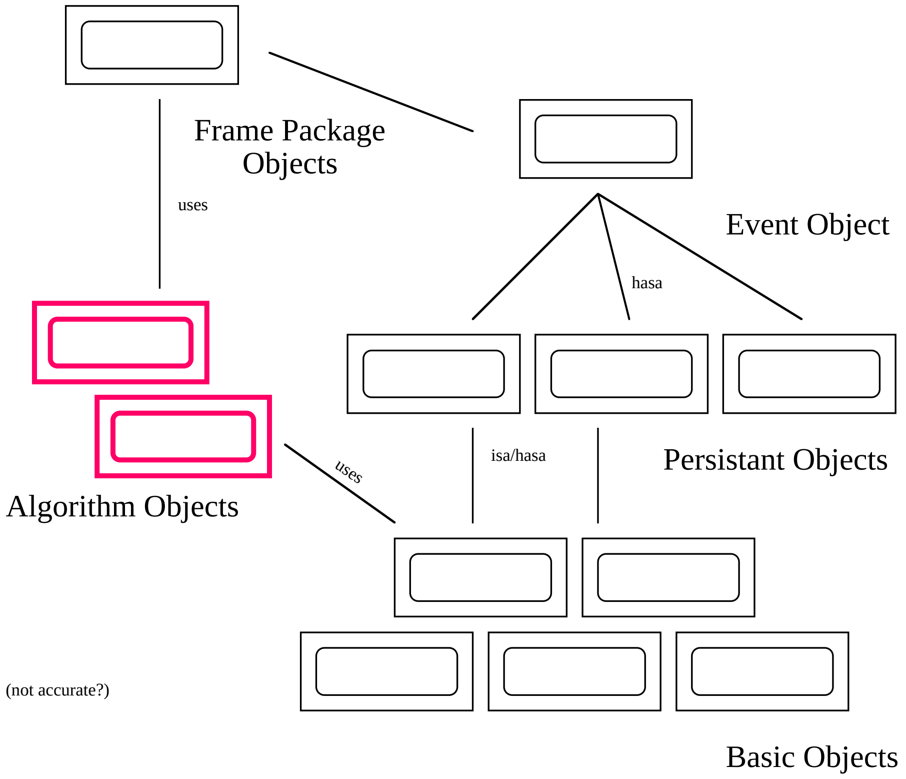
On Demand Reco

On Demand Reconstruction



Assembly Line Reco

Assembly Line Reconstruction



Frame Package Objects

uses

Event Object

hasa

Algorithm Objects

uses

isa/hasa

Persistant Objects

(not accurate?)

Basic Objects

Framework

Environments

- Analysis
  - Beginner
    - Run on a data file.
    - Extract jets/ele.
    - Not get laughed at during talk.
  - Advanced
    - New jet algorithms.
    - Access multiple jets from different algorithms.
    - Overlay different events.
    - Develop code for other environments
- Examines
  - Run from global data or network data
  - GUI that runs in a separate thread and interacts with the algorithms

Work Done

- A generic C++ interface exists
- Physically independent of the scripting system
  - Change to new script system and don't have to recompile
- Can call script objects from C++
- Can call C++ objects from script

Framework

- Level-3 Trigger
  - Light weight, fast, efficient
  - Small blocks of data to be passed to next algorithm
  - Run configuration comes from a database (code must be hot swappable)
  - Accounting!
  - No GUI, program control.
- RECOnstruction
  - No dynamic linking
  - Careful accounting (algorithms, data files)
  - No GUI
  - batch environment

Framework: Run I

Run I used the Assembly Line Model

Several Frameworks:

- Analysis/RECO/Examines
- Level-3
- One hell of a DØOPEN

Development:

- Started in analysis frame
- moved to RECO
- Ported to Level-3
  - Level-3 used the same event model (ZEBRA).

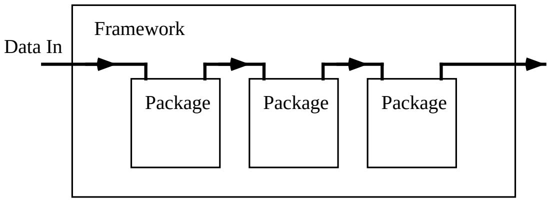
Framework: Run II

- Assembly Line
  - RT data existence checking
  - Only alg requested
  - Alg not associated with data
- On Demand
  - Alg to build data linked in
  - All alg that could be used
  - Alg associated with data

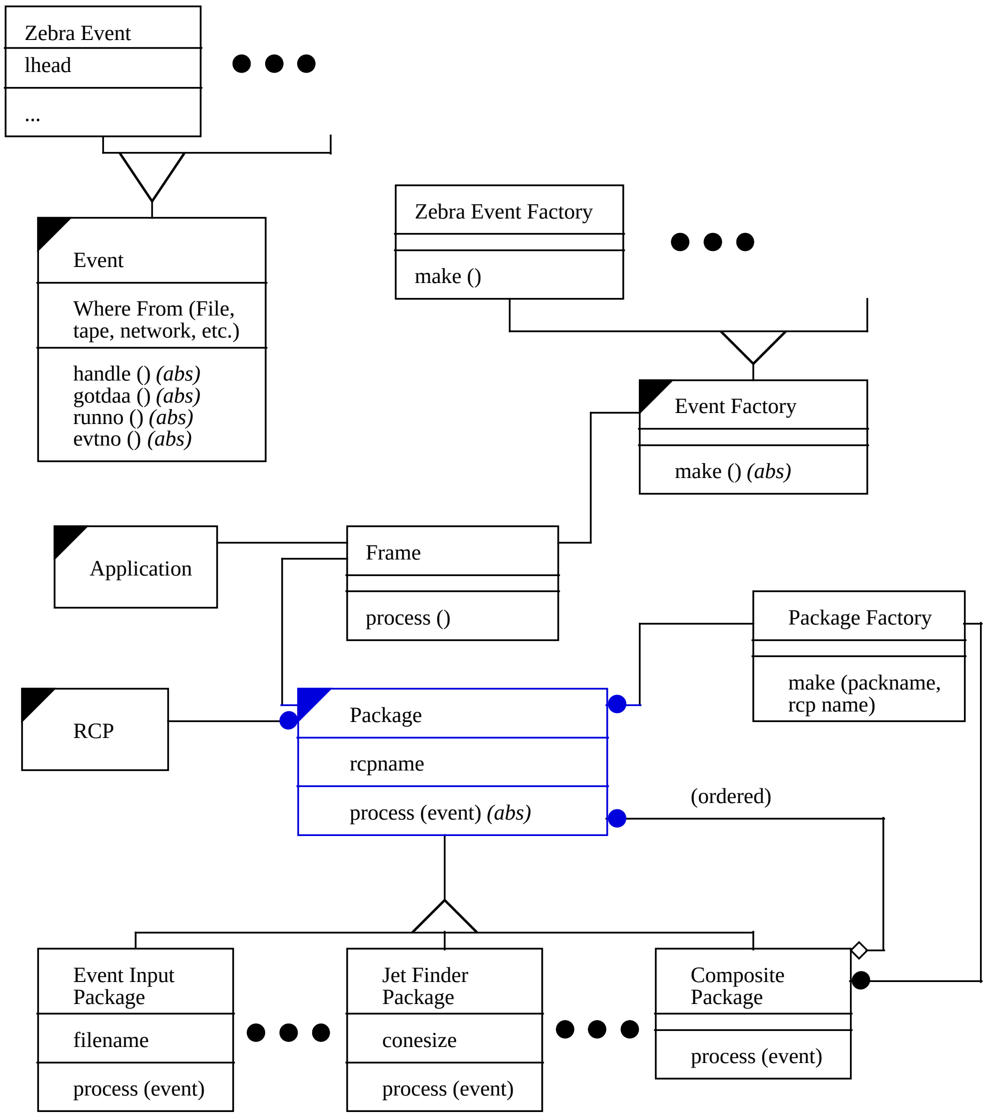
These are extremes!

Use something in between?

Framework: Assembly Line



Framework: Assembly Line



URLs

- **My Home Page:**

  `http://www-d0.fnal.gov/g̃watts/`

- **My Frameworks Page**

  `http://www-d0.fnal.gov/g̃watts/upgrade_software/frame/`

- **There is a link from my frame page to the DØ frame page.**